Local View Based Connectivity Search in Online Social Networks

Na Li
Department of Computer Science
Prairie View A&M University
Prairie View, TX USA
nali@pvamu.edu

Sajal K. Das
Department of Computer Science
Missouri University of Science and Technology
Rolla, MO USA
sdas@mst.edu

***Abstract*—One of the challenges in social media research is that, often times, researchers or third parties could not obtain the massive of data collected by a limited number of "big brothers" (e.g., Facebook and Google). In this paper, we shed light on leveraging social network topological properties and local information to effectively conduct search in Online Social Networks (OSN). The problem we focus on is to discover the reachability of a group of target people in an OSN, particularly from the perspective of a third-party analyst who does not have full access to the OSN. We developed effective and efficient detection techniques which demand only a small number of queries to discover people's connections (e.g. friendship) in the OSN. After conducting experiments on real-world data sets, we found that our proposed techniques perform as well as the centralized detection algorithm, which assumes the availability of the global information in the OSN.**

***Index Terms*—online social networks, subgraph connectivity, search, local view, minimum steiner tree**

I. Introduction

In the past decade, a large number of researchers have shown their interest in Online Social Networks (OSNs). Particularly, they have been dedicated to designing algorithms to solve complex problems relevant to the topological structures of OSN graphs, for example, community detection [1]–[3], detecting subgraphs with a given pattern [4] and sampling social network graphs [5]. Most of those problems assume the availability of entire network graphs, which, however, may not be realistic. Therefore, in recent years, more attention has been paid to leveraging local information and designing distributed algorithms [6]–[10] to solve issues in massive OSNs.

This paper considers the problem of discovering a small subgraph which connects a group of target people in an OSN. We discuss this problem particularly from the perspective of a third-party analyst who does not have a full view of the friendship graph of the OSN site. As a motivating example, consider that a person plans to organize a successful party/workshop, where the success is subject to three constraints: (1) a list of people must be invited to the event (i.e., target people); (2) all target participants should be acquainted with each other directly or through people who need to be invited additionally; and (3) the number of people additionally invited should be minimized due to some reasons, such as budget or space limit.

Solving this problem is challenging for two reasons. First, in an OSN, the information that a third-party analyst can use is limited. The analyst can gather some data either by visiting individual users' profile pages or by sending queries through OSN APIs. What he can see is local to the visited or queried users. Second, even discovering such local information demands effort. One can write script to crawl the web site to collect such data; however, intensively querying the OSN may cause the server to get overwhelmed. This is why many OSNs limit the number of web queries from the same IP address or a particular group of IP addresses per day. Therefore, a third-party analyst needs less-cost searching techniques. The contribution of this paper can be summarized as follows:

- We proposed a subgraph discovery problem in OSNs from the perspective of a third-party analyst.
- We developed technical solutions which integrate some well-known topological properties of social networks to speed the discovery process.
- We conducted experiments on real-world data sets to evaluate the performance of our proposed techniques and found that the reachability of any group of arbitrarily selected nodes could be discovered with a small number of queries in massive OSNs.

The roadmap of this paper is outlined as follows. Section II introduces some preliminaries including the topological properties of social networks, system model and problem definition. Section III and Section IV address the two steps of our proposed searching techniques, online searching and offline detection. Section V discusses the experimental results. Section VI introduces the related work, followed a conclusion in Section VII.

II. Preliminaries

A. Topological properties of Social Networks

Through many years of research in social networks, researchers have detected some important topological properties of social networks after conducting a large number of experiments and analyzing a myriad of real-world data sets.

Small-world Property: It is also translated into "six degrees of separation." It was first observed through a series of experiments conducted by Stanley Milgram and his coworkers in the 1960's [11]–[13]. This property causes the small diameter of social networks and ensures the existence of a short path between any pair of nodes in the social network graphs.

Authorized licensed use limited to: Prairie View A M University. Downloaded on August 05,2023 at 02:43:25 UTC from IEEE Xplore. Restrictions apply.
978-1-7281-1878-9/19/$31.00 ©2019 IEEE

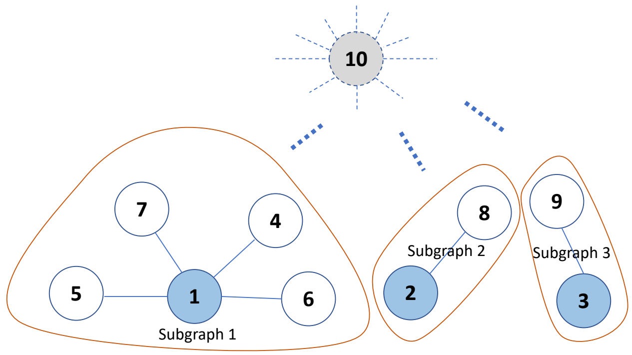


Fig. 1: An example of the search problem

Scale-free Property: A scale-free network has a power-law degree distribution, at least asymptotically, which means only a small number of nodes have very large degrees.

Well-connectivity Among High-degree Nodes: The work [14] has discovered that nodes of high degrees in social network graphs are well connected. In fact, we can prove that even in a random graph, high-degree nodes have higher probability of linking with each other. Our proof is omitted in this paper due to space limit.

B. System Model

Although most OSNs provide all kinds of user relationship information (e.g., friendship or dating relationship) which can be used to measure the acquaintance between people, we consider only friendship and we do not quantify its strength. Therefore, we can use an undirected and unweighted graph, $G(V, E)$, to model the friendship network of an OSN. In the graph model, the node set V represents users and the edge set E denotes the friendship among users.

Then querying a node's friendlist can be modeled as discovering its neighboring friends which we call *local view*. The number of a node's friends is denoted as its degree. When searching for a desired subgraph of friendship with queries, we keep track of not only nodes already queried but also the list of candidates from where the next node for query is chosen. A *candidate* is a node which has been discovered in the query results but has not been queried yet. As more nodes are queried, our view will grow. Figure 1 illustrates an example of a snapshot of an OSN being searched with queries, where blue dots represent nodes already queried, white ones are candidate nodes, and the gray node represents the one which exists in the OSN but has not been discovered yet.

C. Problem Definition

We define Local-view based Minimum Subgraph Detection problem (LMSD) in Problem 1. The LMSD problem requires both the size of the detected subgraph in terms of the number of nodes and the number of queries be minimized, which, however, is hard to achieve at the same time. To cope with this challenge, we heuristically interpret the problem and break it down to two steps. We first conduct online search with a small number of queries to find a connected subgraph covering all target nodes, and then in the detected subgraph we aim to discover the subgraph which keeps all target nodes connected with the minimum number of nodes involved. The rationale of the effectiveness of our interpretation in solving the LMSD problem is if the number of nodes queried in the first step is small, the size of the finally detected subgraph should not be very large.

Problem 1 *Local-view based Minimum Subgraph Detection (LMSD): Given a set of target nodes S_0 in a graph, $G(V, E)$, the full topology of which is unknown initially, find the minimum number of nodes from $V \setminus S_0$ to make all target nodes connected with the minimum number of node queries for local-view discovery.*

Given the subgraph discovered in the first step, we name the minimum subgraph detection problem in the second step the Centralized Minimum Subgraph Detection problem (CMSD): given a graph and a group of target nodes, we look for the minimum number of extra nodes to connect all of the target nodes together. The CMSD problem is a hard problem as proved in Theorem 1. The complexity of the CMSD indirectly indicates the hardness of the LMSD problem. Based on our two-step based interpretation, we will first discuss how to detect the connectivity of target nodes via a small number of queries in Section III and then address how to discover the reachability of target nodes offline through a small number of extra nodes in the subgraph from the previous step.

Theorem 1 *The Centralized Minimum Subgraph Detection problem (CMSD) is NP-hard.* □

PROOF We will prove the NP-hardness of CMSD by a reduction to the Steiner Tree problem (ST). The definition of ST is: Given an unweighted graph G and a set of nodes V_t in it, find a tree with minimum number of edges in G, which make any two nodes in V_t reachable to each other either directly or indirectly via other nodes in G. As is well known, the ST problem is NP-hard [15]. The decision version of ST is that given an unweighted graph $G(V, E)$, a set of nodes $V_t \subseteq V$ and an integer k, we are looking for a tree which involves all nodes in V_t and contains at most k edges from E. The decision version of CMSD problem is that given an unweighted graph $G'(V', E')$, a set of nodes $V'_t \subseteq V'$ and an integer k', we are searching for a subgraph of G' which includes all nodes in V'_t and covers at most k' nodes from $V' \setminus V'_t$.

We can demonstrate that there is a solution for ST if and only if there is a solution for CMSD. Evidently, the nodes in any steiner tree with at most k edges will be the solution of CMSD, where $k' = k + 1 - |V_t|$. On the other hand, any spanning tree of the subgraph found in CMSD can form a steiner tree with at most $k' + |V'_t| - 1$ edges. Here the spanning tree is referred to as a tree composed of all the nodes and some (or perhaps all) of the edges of a given graph. Therefore, the CMSD problem is NP-Hard. ■

III. Online Searching

In the online searching, we search for a subgraph to connect all target nodes in the OSN with a small number of queries. The traditional graph searching techniques, such as *Depth First*

Authorized licensed use limited to: Prairie View A M University. Downloaded on August 05,2023 at 02:43:25 UTC from IEEE Xplore. Restrictions apply.

Search (DFS) or *Breadth First Search* (BFS), can be applied as the brute-force subgraph detection techniques; however, their cost on queries for location view discovery is non-trivial, due to the lack of global topology information of the OSN. Therefore, we are motivated to design more efficient searching techniques to discover the connectivity of targets.

A. The starting point of search

Without any prior knowledge, the search should start from the target nodes. After all target nodes are queried, each of them and its neighbors discovered through the OSN API form a subgraph, as illustrated in Figure 1, where nodes 1, 2 and 3, are targets. These subgraphs are most likely disjoint due to the structural sparsity of social networks. Each of these subgraphs has its own node candidate set. The candidate set of a subgraph initially contains only the neighbors of its target node, but it grows as more nodes are queried.

Given the scattered subgraphs, efficiently discovering the connectivity of target nodes requires merging all of these subgraphs quickly by querying a small number of nodes. Therefore, the selection of nodes for query is critical. In the following subsections, we will discuss how to evaluate the importance of a node in the online searching.

B. The evaluation of node candidates

In a dense graph ($|E| \gg |V|$), it's straightforward to pick a good node candidate for query. Basically, a node which can make more target nodes accessible to each other should be selected. However, such a criterion is not sufficient to determine a candidate node in a sparse graph, such as social network graphs, and may even lead to the failure of the search process. The reason is that in a sparse graph, more likely, none of the node candidates can directly improve the reachability of target nodes. Therefore, we need a new criterion to evaluate a node's capability of facilitating the merging of subgraphs.

Inspired by the critical role of high-degree nodes in searching in social network graphs [16], [17], we prioritize node candidates of high degrees for query in the online searching. However, often times, the real degree of a node candidate is unknown until we query it, and thus, we estimate node degree based on the discovered information to reduce query cost. Specifically, we count the number of a node's connections alredy discovered in the online searching upon to a time point and use it to estimate the node's degree, which we call *Pre-Degree*. In Figure 1, the pre-degree of node 9 at that time point is one, as we only see its connection with node 3. As more nodes are queried, we may discover more connections with node 9 which will cause its pre-degree to increase accordingly. The rationale of estimating a node degree with its pre-degree is that since social network graphs have power law degree distributions, if we see a node has a high pre-degree, the real degree of that node will probably be high as well. Of course, this may not always be accurate when a node's pre-degree is low.

In fact, we also explored to use some other information than pre-degree to estimate node real degree, such as user account creation time. We ran some experiments to compare it with the pre-degree and real degree startegies; however, the discussion is omitted in this paper due to space limit.

C. Algorithmic Techniques for Online Searching

We propose two online search techniques, called *Unbalanced Multiple-Subgraph Searching (UMS)*, and *Balanced Multiple-Subgraph Searching (BMS)*, respectively. The two techniques have similar algorithmic logic, as illustrated in Algorithm 1. The techniques consist of three steps: (1) query all of the target nodes in the OSN graph and form their individual subgraphs; (2) select a subgraph as the *target subgraph*; and (3) query the candidate node of the largest estimated degree (e.g., pre-degree) in the target subgraph. A tie will be broken arbitrarily. If a query discovers any overlap of the subgraphs, then merge them as one. The last two steps will be repeated until all subgraphs of target nodes are merged together.

The difference between UMS and BMS is how to select target subgraph. In order to evaluate a subgraph, we define *subgraph degree* as the maximum (estimated) degree of nodes in the subgraph. UMS picks the subgraph with maximum subgraph degree, while BMS selects the subgraph with minimum subgraph degree. In Figure 1, after querying the target nodes, 1, 2 and 3, individually, we get three disjoint subgraphs which have degrees of 4, 1 and 1, respectively. With UMS, Subgraph 1 is assigned as the target subgraph and then node 4 is queried (break tie arbitrarily). For UMS, once a target subgraph is determined, it will not be reassigned. This is because a query upon the target subgraph will lead to the discovery of more edges so that its subgraph degree will increase, thereby keeping the maximum degree among all subgraphs. However, for BMS, once a query is made, the current target subgraph may not have the minimum degree any more. Let's continue to use the example illustrated in Figure 1. With BMS, Subgraph 2 or 3 (break tie arbitrarily) will be selected as the target as they have a low degree of 1.

The inspiration in designing BMS comes from our concern over the efficiency of searching with UMS. One can see that essentially UMS prioritizes high-degree nodes in the search, which may not be able to efficiently reach the target nodes of low degrees. As illustrated in Figure 1, with UMS, Subgraphs 2 and 3 have to wait until the search reaches them. We believe since the high-degree nodes in social networks are well connected as we introduced in Section II, if we could reduce the degree difference among the subgraphs by prioritizing the subgraphs of low degrees in searching, the procedure of merging subgraphs may perform faster. Therefore, we develop this technique called Balanced Multiple-Subgraph Searching.

IV. Offline Detection

In the offline detection phase, we aim to find a smaller subgraph from the subgraph discovered in the online searching which can maintain the connectivity of all target nodes. Considering the association between the CMSD problem and the Steiner Tree problem (ST) as we discussed in Section II,

Authorized licensed use limited to: Prairie View A M University. Downloaded on August 05,2023 at 02:43:25 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1: The Framework of Our Online Searching Techniques

Input: Set of target nodes, TS, and an oracle of querying nodes in the OSN
Output: A connected subgraph covering all nodes in TS
foreach *Node* v_i *in* TS **do**
  subgraph(i) = Query(v_i);
  $list$.add(subgraph(i));
while $list.size$ *!= 1* **do**
  tsg = SelectTargetSubgraph($list$);
  tn = SelectNode(tsg);
  subgraph(tsg).add(Query(tn));
  if *CheckOverlap(*$list$*,* tsg*)* **then**
    Merge($list$, tsg);
  foreach *Node candidate* v_m *in* tsg **do**
    Update($D(v_m)$);

we apply a classic approximate ST algorithm [18] to detect a smaller subgraph in the offline detection phase.

There are two main reasons for us to apply the ST algorithm [18]. First, it can guarantee the size of the detected subgraph is no larger than $2(1 - 1/\ell)$ times the size of the optimal subgraph, where ℓ is the number of leaves in the optimal tree. Second, it runs faster with the time complexity $|S_0||V|^2$, which is a critical concern when running algorithms on large-scale OSN data sets.

Given an undirected and unweighted graph $G(V, E)$ and a set of target nodes $S_0 \subseteq V$, there are four steps to find a heuristic minimum steiner tree in [18]: (1) construct the complete undirected graph $G_1(V_1, E_1)$ by creating an edge between each pair of nodes in S_0 with a label of the length of their shortest path on G; (2) find the minimal spanning tree T_1 of G_1; (3) construct a subgraph G_s of G by replacing each edge in T_1 by its corresponding shortest path in G; and (4) find the minimal spanning tree T_s of G_s. Delete from T_s edges with leaves which are non-steiner points.

V. Experimental Study

To evaluate the performance of our techniques in solving the LMSD problem, we conducted experiments on large-scale real-world data sets. In this section, we will first introduce the data sets used in our experiments and analyze their topological properties, including the degree distribution and the connectivity of high-degree nodes. Then, we will evaluate the two steps of our online searching techniques, node selection and target subgraph selection.

A. Data Sets

We used two real-world data sets in our experiments, which can be downloaded from the repository [19] .

(1) Gowalla data set [20]: Gowalla is a location-based social networking website where users share their locations by checking in. The data collected from Gowalla present the friendship network which is undirected.

TABLE I: Subgraphs of high-degree nodes

$\geq$ Degree		100	200	300	400	500	600
Gowalla	**nodes**	1787	494	245	143	99	77
	Comps	1	1	1	1	1	1
	Avg. Dist.	2.32	1.96	1.82	1.70	1.63	1.58
Brightkite	**nodes**	408	89	30	14	9	7
	Comps	1	1	1	1	1	1
	Avg. Dist.	2.29	1.95	1.88	1.78	1.67	1.71

(2) Brightkite data set [20]: Brightkite was once a location-based social networking service provider where users shared their locations by checking-in. The friendship network is originally directed but we have constructed a network with undirected edges whenever there is a friendship regardless of the direction.

As we study on how to connect a group of nodes together on an OSN, we need to ensure all of the target nodes are indeed reachable to each other in the data sets, which means the undirected input graph of each data set should be connected. Therefore, we processed the original data sets by extracting their largest connected components. The largest components still retain a large number of nodes and edges: 196591 and 950327 in Gowalla; and 56739 and 212945 in Brightkite.

B. Topological Properties of Data Sets

We examine some of the topological properties of our data sets which we introduce in Section II, including power-law degree distribution and the well connectivity of high-degree nodes, .

1) Power-law degree distribution: We apply the statistical framework for discerning and quantifying power-law behavior in empirical data proposed in [21] to check the degree distribution of our data sets. The framework code is available at [22]. In power-law distribution, $P(x) \sim x^{-\alpha}$, α is known as the exponent or scaling parameter, which typically lies in the range $2 < \alpha < 3$. More often the power law applies only for values greater than some minimum $xmin$. In such cases we usually say that the tail of the distribution follows a power law. Therefore, we check the values of the two parameters, α and $xmin$, for our data sets: for Gowalla, α = 2.83 and $xmin$ = 95; and for Brightkite, α = 2.56 and $xmin$ = 24.

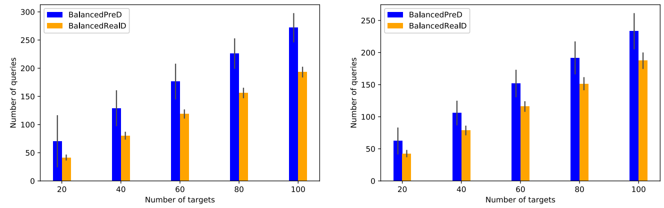
2) Connectivity of High-Degree Nodes: To evaluate the connectivity of high-degree nodes in our data sets, we first extracted the subgraph consisting of nodes with a degree more than a certain threshold and their edges. The threshold ranges from 100 to 600 in increments of 100. We analyze the number of the connected components and the average distance of shortest paths between any pair of nodes for each data set.

From Table I, we can see that although the number of nodes decreases as the degree threshold goes up, the nodes of high degree are still connected well. Also, we can see the average distance between any pair of reachable nodes is about 2. These results demonstrate the well-connectivity among high-degree nodes in our OSN data sets.

C. The Evaluation of Techniques

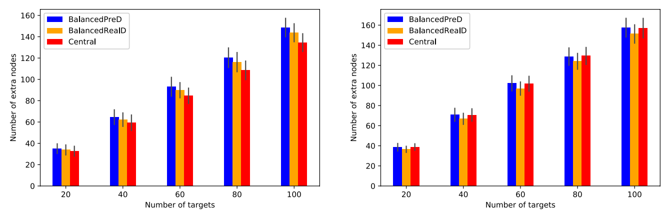
We conduct multiple groups of experiments for each data set by varying the number of target nodes, ranging from 20 to 100 in increments of 20. Given a specific number of target nodes,

Authorized licensed use limited to: Prairie View A M University. Downloaded on August 05,2023 at 02:43:25 UTC from IEEE Xplore. Restrictions apply.



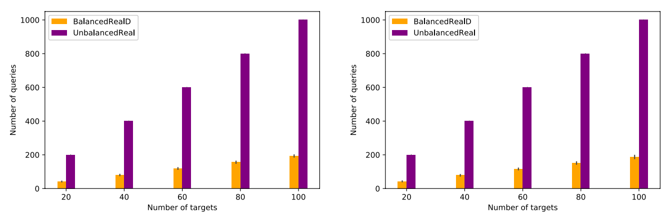
(a) (b)

Fig. 2: The number of queries with different node selection strategies. (a) Gowalla; (b) Brightkite.



(a) (b)

Fig. 3: The number of extra nodes with different node selection strategies. (a) Gowalla; (b) Brightkite.



(a) (b)

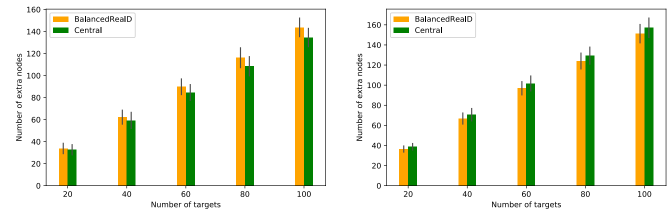
Fig. 4: The number of queries with BMS and UMS. (a) Gowalla; (b) Brightkite.

we run the experiment 100 times, each time selecting target nodes uniformly at random from each data set. As discussed in Section III, there are two steps in the online searching, choosing the target subgraph first and then selecting a node to query. Therefore, we evaluate these two steps, respectively.

1) Node selection (NS): In this group of experiments, we evaluate the pre-degree based node selection by comparing it with the node selection of using their real degrees. We validate whether high-degree nodes are good choice for search. Also, we aim to verify the goodness of using pre-degree to estimate the real degree of a node candidate. In the experiments, along with the node selection techniques, we use the balanced subgraph selection, which targets the subgraph with the lowest degree. We evaluate node selection techniques in terms of the number of online queries and the number of extra nodes selected in the offline to make the target nodes reachable.

In terms of the number of queries, in Figures 2a-2b, we can see: (1) on the average, the number of queries issued with the pre-degree based node selection (BalancedPreD) is more than that number with the node selection of using their real degrees (BalancedRealD). (2) BalancedPreD needs less queries than 2 times of the number of targets.

In terms of the number of extra nodes needed to connect the targets, in Figures 3a-3b, the performances of BalancedPreD, and BalancedRealD are quite similar to that of the centralized algorithm (Central) which runs the MST on the entire input data set. It demonstrates that with the local view in the social network graph we can still efficiently discover the connectivity of a group of target nodes in a massive OSN.



(a) (b)

Fig. 5: The number of extra nodes with BMS and UMS. (a) Gowalla; (b) Brightkite.

2) Subgraph Selection (SS): In the experiments of evaluating subgraph selection techniques, we test the unbalanced subgraph selection (UMS), and the balanced subgraph selection (BMS). The UMS always targets the subgraph with the largest degree, while the BMS always chooses the subgraph with lowest degree. In this group of experiments, we apply the real degree to node selection aiming at eliminating the impact of node selection on evaluating the performance of subgraph selection techniques. After running some experiments, we realized that the number of queries issued with BMS is about two times of the number of target nodes; however, the UMS may search through a large portion of the data set. Therefore, for the UMS, we set a termination condition: if the number of queries issued is more than 10 times of the number of targets, we will terminate the search.

In Figures 4a-4b, we can see that the BalancedRealD uses a fewer number of queries to discover the connectivity of target nodes. However, in most of the cases, the UnbalancedReal was terminated by the condition we set, which means the number of queries issued with UMS is more than 10 times of the number of the target nodes.

In terms of the extra nodes needed for connecting the targets offline, since the connectivity wasn't achieved, we only compare BalancedRealD with Central. Their performance is quite similar, as displayed in Figures 5a-5b.

VI. Related Work

A. Subgraph Connectivity

Our subgraph detection problem is relevant to the subgraph connectivity in the domain of graph mining. [23] proposes solutions for finding a subgraph that connects a set of query nodes in a graph, where the proximity between nodes is defined depending on the global topology of the graph. [24] addressed the searching for the densest community containing all query nodes with and without size constraint. Most recently,

Authorized licensed use limited to: Prairie View A M University. Downloaded on August 05,2023 at 02:43:25 UTC from IEEE Xplore. Restrictions apply.

[25] examines the Steiner Maximum-Connected Subgraph (SMCS) problem. The main difference between our proposed problem and the above line of research is two-fold: (1) The existing work addresses subgraph connectivity with pre-known global topology; however, we consider the subgraph detection by a third-party analyst. (2) Our problem is different as we consider small subgraph connectivity as well as query cost.

B. Local View Based Graph Algorithms

[9] proposes local graph clustering methods to find a cluster of nodes by exploring a small region of the graph, which enable targeted clustering around a given seed node and are faster than traditional global graph clustering methods because their runtime does not depend on the size of the input graph. Additionally, [8] proposes a local search in the neighborhood of a node to find the best community for the node. The difference between our work and the above work is that we consider a different search problem, and also, we particularly take advantage of topological properties of social networks in the design of search strategies.

VII. Conclusion

In this paper, we propose a problem of discovering a minimum subgraph covering a given group of nodes from the perspective of third-party analysts in OSNs, namely local-view based minimum subgraph detection (LMSD). To solve this problem, we propose two searching techniques, called Unbalanced Multiple-Subgraph (UMS) and Balanced Multiple-Subgraph (BMS), which are based on the well-known topological properties of social networks, including small-world phenomenon, power-law node degree distribution and the well-connectivity of nodes of high degrees.

Through experiments over large-scale real-world data sets, we evaluate the performance of our proposed techniques. The BMS technique performs better than UMS, which demonstrates that the well-connectivity property in social networks is not restricted to nodes of high degrees in OSNs, rather, the entire OSNs are well connected, as any group of arbitrarily selected nodes can reach connectivity by a small number of node queries. Furthermore, the design principle in BMS of searching from low-degree subgraphs shows great impact on the efficiency in solving the LMSD problem. Our work sheds light on leveraging social network topological properties to conduct search efficiently, which may improve some of the existing searching-related research work in OSNs.

Acknowledgement

This project is supported in part by the National Science Foundation (NSF) under grants DUE-1712496, CCF-1725755 and CCF-1533918. Any opinions, findings, and conclusions expressed in this paper are those of the authors, and do not necessarily reflect the views of NSF.

References

[1] P. Bedi and C. Sharma, "Community detection in social networks," *Wiley Interdisciplinary Reviews Data Mining and Knowledge Discovery*, pp. 496–500, 2016.

[2] M. Wang, C. Wang, J. X. Yu, and J. Zhang, "Community detection in social networks: an in-depth benchmarking study with a procedure-oriented framework," *VLDB*, vol. 8, no. 10, pp. 998–1009, 2015.

[3] L. Qin, J. X. Yu, and R. Mao, "Influential community search in large networks," *VLDB Endowment*, vol. 8, no. 5, pp. 509–520, January 2015.

[4] W. Fang, "Graph pattern matching revised for social network analysis," in *Proceedings of the 15th International Conference on Database Theory*, 2012, pp. 8–21.

[5] M. Papagelis, G. Das, and N. Koudas, "Sampling online social networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 3, pp. 662 – 676, 2013.

[6] X. Zhao, A. Sala, H. Zheng, and B. Y. Zhao, "Efficient shortest paths on massive social graphs," in *Proceedings of IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*, 2011, p. 7786.

[7] C. Borgs, M. Brautbar, J. Chayes, S. Khanna, and B. Lucier, "The power of local information in social networks," *Internet and Network Economics, Springer*, vol. 7695, p. 406419, 2012.

[8] W. Cui, Y. Xiao, H. Wang, and W. Wang, "Local search of communities in large graphs," in *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, June 2014, pp. 991–1002.

[9] H. Yin, A. R. Benson, J. Leskovec, and D. F. Gleich, "Local higher-order graph clustering," in *Proceedings of ACM SIGKDD*, 2017, pp. 555–564.

[10] P. Basuchowdhuri, S. Sikdar, V. Nagarajan, K. Mishra, S. Gupta, and S. Majumder, "Fast detection of community structures using graph traversal in social networks," *Knowledge and Information Systems*, pp. 1 – 31, May 2018.

[11] C. Korte and S. Milgram, "Acquaintance networks between racial groups: Application of the samll world method," *Journal of Personality and Social Psychology*, vol. 15, no. 2, pp. 101–108, 1970.

[12] S. Milgram, "The small world problem," *Psychology Today*, vol. 1, no. 61, pp. 60–67, 1967.

[13] J. Travers and S. Milgram, "An experimental study of the samll world problem," *Sociometry*, vol. 32, no. 425, pp. 425–443, 1969.

[14] C. Wilson, B. Boe, A. Sala, K. P. Puttaswamy, and B. Y. Zhao, "User interactions in social networks and their implications," in *Proceedings of EuroSys*, 2009.

[15] M. Garey and D. Johnson, *Computers and Intractability*. Freeman, 1979.

[16] L. Adamic and E. Adar, "How to search a social network," *Social Networks*, vol. 27, no. 3, pp. 187–203, 2005.

[17] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman, "Search in power-law networks," *Physical Review*, vol. 64, no. 4, 2001.

[18] L. Kou, G. Markowsky, and L. Berman, "A fast algorithm for steiner trees," *Acta Informatica*, vol. 15, no. 2, pp. 141–145, 1981.

[19] "Stanford Large Network Dataset Collection," http://snap.stanford.edu/data/, 2019, [Online; accessed 14-March-2019].

[20] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: Friendship and mobility: User movement in location-based social networks," in *Proceedings of ACM SIGKDD*, 2011, pp. 1082–1090.

[21] A. Clauset, C. R. Shalizi, and M. E. J. Newman, "Power-law distributions in empirical data," *Society for Industrial and Applied Mathematics*, vol. 51, no. 4, pp. 661–703, 2009.

[22] A. Clauset, "Fitting a power-law distribution," http://tuvalu.santafe.edu/~aaronc/powerlaws/, 2019, [Online; accessed 14-March-2019].

[23] H. Tong and C. Faloutsos, "Center-piece subgraphs: problem definition and fast solutions," in *Proceedings of SIGKDD*, 2006, pp. 404–413.

[24] M. Sozio and A. Gionis, "The community-search problem and how to plan a successful cocktail party," in *Proceedings of ACM SIGKDD*, 2010.

[25] J. Hu, X. Wu, R. Cheng, S. Luo, and Y. Fang, "On minimal steiner maximum-connected subgraph queries," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 11, pp. 2455 – 2469, 2017.

Authorized licensed use limited to: Prairie View A M University. Downloaded on August 05,2023 at 02:43:25 UTC from IEEE Xplore. Restrictions apply.